Multivariate Dynamic Kernels for Financial Time Series Forecasting

Mauricio Peña, Argimiro Arratia, and Lluís A. Belanche

Dept. of Computer Science**
Technical University of Catalonia, Jordi Girona, 1-3 08034, Barcelona, SPAIN
mpenagrass@gmail.com,argimiro@cs.upc.edu,belanche@cs.upc.edu

Abstract. We propose a forecasting procedure based on multivariate dynamic kernels, with the capability of integrating information measured at different frequencies and at irregular time intervals in financial markets. A data compression process redefines the original financial time series into temporal data blocks, analyzing the temporal information of multiple time intervals. The analysis is done through multivariate dynamic kernels within support vector regression. We also propose two kernels for financial time series that are computationally efficient without a sacrifice on accuracy. The efficacy of the methodology is demonstrated by empirical experiments on forecasting the challenging S&P500 market.

Keywords: support vector regression; financial time series; kernels

1 Introduction

The forecasting of financial markets is one of the most challenging tasks in predictive analytics. The non-stationarity and the noisy nature of financial time series have driven the debate about whether it is really possible to predict market movements with sufficient confidence. The "Efficient Market Hypothesis" provides theoretical grounds for the belief that the best strategy is the "buy-and-hold" passive investment strategy, since no excess return can be obtained consistently by predicting and timing the market [1].

Although many researchers in the statistical learning community –see e.g. [2–4]– have attempted to forecast the financial market using support vector machines (SVM) with standard kernels, the area still remains a challenge for practitioners. Therefore, there is a natural interest in applying kernels for financial forecasting by incorporating temporal information between misaligned time series or varying frequencies in the data patterns. In this article, we propose a forecasting methodology based on SVMs that permits the incorporation of granular temporal information of variable-length time series. The proposed forecasting methodology is a very flexible approach capable of analyzing market dynamics in very short-term intervals, by integrating market micro-structure information in a compressed fashion. Standard kernels in the literature are replaced

** Supported by MINECO project APCOM (TIN2014-57226-P) and *Generalitat de Catalunya* 2014 SGR 890 (MACDA).

by *dynamic* kernel functions able to analyze multivariate temporal structures. We show how the use of these kernels leads to improvements in terms of both accuracy and forecasting performance. In addition, we propose some multivariate dynamic kernels that make it possible to reduce the complexity of kernel analytics to a manageable level without compromising on accuracy. The computational speed of these kernels makes them ideal candidates for intensive computational tasks. The approach can be extended to incorporate high-frequency information as well, aimed at market risk measurement.

2 Preliminaries

Support Vector Machines for Regression. We use Support Vector Regression (SVR) for predicting one-month ahead market performance by using its own history and a series of exogenous variables measured on a daily basis; thus, it is a mixed-frequency approach. More specifically, we choose the ν-SVR, a reformulation that involves the automatic adaptation of the ϵ parameter. The ν parameter is bounded in the interval $(0, 1]$, representing both an upper bound on the fraction of training samples which are errors and lower bound on the fraction of points which are support vectors [5]. The final dual expression for an SVR is

$$y_{\mathrm{SVM}}(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(x, x_i)$$

where α_i, α_i^* are the dual variables $(0 \leq \alpha_i, \alpha_i^* \leq C)$, $C > 0$ is the regularization parameter, the $\{x_i\}$ are the training points, and k is the kernel function.

Data Blocks for Temporal Information. Practitioners usually apply time series regression with SVR using standard static kernels such as the Gaussian, linear and polynomial. This means that, for one-month ahead predictions, there is only a single vector of prices for each input month. To extract additional information and incorporate more subtle patterns, we propose that daily quotes of financial assets be compressed into temporal time intervals on each month. Our compression process redefines the original dataset into new instances $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_j, \ldots$ taking the form of *multivariate time series* (MVT), as described next. A univariate time series $x_i = \{x_i(1), x_i(2), \ldots, x_i(T_j)\} \in \mathbb{R}^{T_j}$ of length T_j is a set of observations from a random process measured at discrete intervals of time. The j-th MVT is then a P-by-T_j matrix $\boldsymbol{X}_j \in \mathbb{R}^{P \times T_j}$ of the form

$$\boldsymbol{X}_j = \left(\begin{bmatrix} x_1(1) \\ x_2(1) \\ \vdots \\ x_P(1) \end{bmatrix} \ldots \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_P(t) \end{bmatrix} \ldots \begin{bmatrix} x_1(T_j) \\ x_2(T_j) \\ \vdots \\ x_P(T_j) \end{bmatrix} \right) \tag{1}$$

where each row represents a univariate time series and each column is a vector of observations of the P variables in a time point. Letting $x(i)$ be the i-th column of $\boldsymbol{X}_j (i = 1, \ldots, T_j)$, the MVT $\boldsymbol{X}_j$ can be expressed as $\boldsymbol{X}_j = (x(1), \ldots, x(T_j))$.

Therefore, the original dataset is transformed into several intervals of different sizes where each instance is now expressed as in Eq. (1). This allows to model the temporal structure within months and, additionally, can be adapted to incorporate market dynamics in very small time intervals.

3 Multivariate Dynamic Kernels

The general goal is to define positive definite (p.d.) kernels between two time series (not necessarily of the same length), $\boldsymbol{X} = (x(1), \ldots, x(N))$ and $\boldsymbol{Y} = (y(1), \ldots, y(M))$, where the pairwise comparisons $(x(i), y(j))$ are reasonable. The main difficulty is that the commonly used Euclidean distance disregards the temporal dependency among the observations of time series. Moreover, the length of the different time series is variable since it is a function of the number of business days of each month, among other causes. In an attempt to overcome the aforementioned difficulties, Sakoe and Chiba proposed *dynamic time warping* (DTW), to find a good alignment between $\boldsymbol{X}$ and $\boldsymbol{Y}$ before computing any Euclidean distance [6]. An *alignment* (or *warping function*) π between two time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a pair of increasing tuples (π_1, π_2) of length $P \leq N + M - 1$ such that $1 = \pi_1(1) \leq \ldots \leq \pi_1(P) = N$ and $1 = \pi_2(1) \leq \ldots \leq \pi_2(P) = M$, with unitary increments and no simultaneous repetitions. Intuitively, an alignment is a series of connecting lines that associate each time point of $\boldsymbol{X}$ to one or more time points in $\boldsymbol{Y}$, and vice versa, as:

$$D_\pi(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{|\pi|} \|x_{\pi_1(i)} - y_{\pi_2(i)}\|^2$$

The *multivariate dynamic time warping* (MDTW) distance is the minimum distance for the set of all alignments $\mathrm{AL}(\boldsymbol{X}, \boldsymbol{Y})$:

$$\mathrm{MDTW}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{|\pi^*|} \min_{\pi \in \mathrm{AL}(\boldsymbol{X}, \boldsymbol{Y})} D_\pi(\boldsymbol{X}, \boldsymbol{Y}), \text{ with } \pi^* = \operatorname*{arg\,min}_{\pi \in \mathrm{AL}(\boldsymbol{X}, \boldsymbol{Y})} D_\pi(\boldsymbol{X}, \boldsymbol{Y}).$$

To convert a MDTW distance into a similarity we use the Gaussian function with parameter $\sigma > 0$ as $k_{\mathrm{MDTW}}(\boldsymbol{X}, \boldsymbol{Y}) = \exp(-\mathrm{MDTW}(\boldsymbol{X}, \boldsymbol{Y})/\sigma)$. The main drawback of the DTW measure is that it is not rigorously a metric (it does not satisfy the triangle inequality) and is also known *not* to be conditionally n.d.; hence its negative exponential is not a p.d. kernel in general. Moreover, since the DTW is based exclusively on the optimal alignment π^*, counter-intuitive behaviors can be obtained in some cases –see [7].

Global Alignment Kernel. In view of the limitations of the DTW, we consider an improvement given by the *global alignment* (GA) kernel [7], which instead of the minimum it considers the *soft-minimum* of $D_\pi(\boldsymbol{X},\boldsymbol{Y})$ defined as

$$\mathrm{Smin}(D_\pi(\boldsymbol{X},\boldsymbol{Y})) = -\log \sum_{\pi\in \mathrm{AL}(\boldsymbol{X},\boldsymbol{Y})} e^{-D_\pi(\boldsymbol{X},\boldsymbol{Y})}$$

To get a kernel, take $\exp(-\mathrm{Smin}/\sigma)$ as $k_{\mathrm{GA}}(\boldsymbol{X},\boldsymbol{Y}) = \sum_{\pi\in \mathrm{AL}(\boldsymbol{X},\boldsymbol{Y})} e^{-D_\pi(\boldsymbol{X},\boldsymbol{Y})/\sigma}$.

The GA kernel takes advantage of the distances spanned by all possible alignments: two time series are similar based on their *set* of efficient alignments. The σ parameter is taken from the adaptative grid:

$$\{0.2, 0.4, \ldots, 2\} \cdot \mathrm{median}(\|x(t_1) - y(t_2)\|) \cdot \sqrt{\mathrm{median}(|x(t_1)|)},$$

where $x(t_1)$ and $y(t_2)$ are time points for the days in which the target price reached its minimum variation during the month of each time series.

Vector Autoregressive Kernel. The previous kernels are shape-based similarities to compare two time series. In this work we also propose the extraction of higher-level dependencies across time series through a parametric statistical model. Our approach, a straightforward adaptation of the VAR kernel [8], is based on comparing the similarity of two time series using the transition matrices and intercepts of a *vector autoregressive model* VAR(L), such that $x(t) = \sum_{l=1}^{L} A_l x(t-l) + b + \varepsilon_t$, where $A_1, \ldots, A_L \in \mathbb{R}^{P\times P}$ are the transition matrices, $b \in \mathbb{R}^P$ is the intercept, and $\varepsilon \sim \mathcal{N}(0, \Sigma)$ is the noise. To implement the VAR kernel, we append the estimated parameters $\hat{A}$ and $\hat{b}$ into a single matrix $\hat{B} = (\hat{A}_1|\hat{A}_2|\ldots|\hat{A}_L|[\hat{b}])$. and then compute a distance between time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ using the Frobenius norm over the difference of their $\hat{B}$ matrices

$$\mathrm{FD}(\boldsymbol{X},\boldsymbol{Y}) = \sqrt{\mathrm{Trace}\Big\{(\hat{B}_{\boldsymbol{X}} - \hat{B}_{\boldsymbol{Y}})(\hat{B}_{\boldsymbol{X}} - \hat{B}_{\boldsymbol{Y}})^T\Big\}}$$

To convert the Frobenius distance to a similarity measure, we use a Gaussian function to get $k_{\mathrm{VAR}}(\boldsymbol{X},\boldsymbol{Y}) = \exp(-\mathrm{FD}(\boldsymbol{X},\boldsymbol{Y})/\sigma)$. For the experiments, we use a fixed lag of $L = 5$ as indicated in [8] and set σ as the median Frobenius distance.

Multivariate Dynamic Euclidean Distance Kernel. Finally we propose a simple but effective methodology to compare variable-length time series by constructing what we call the *multivariate dynamic euclidean distance* (MDED) kernel. Given that financial time series follow a filtration process, we propose an alignment that shortens the longer time series so to become equal in length to the shorter one. Formally, the MDED alignment between time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ with respective lengths $N \geq M$ is $\pi_{\mathrm{MDED}} = \{(N-(M-1), 1), (N-(M-2), 2), \ldots, (N-1, M-1), (N, M)\}$. We then define the multivariate dynamic Euclidean distance as $\mathrm{MDED}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{M}\sum_{i=1}^{M} \|(x_{\pi_{\mathrm{MDED}}(i,1)} - y_{\pi_{\mathrm{MDED}}(i,2)}\|^2$. These distances can be fairly compared across variable-length time series in the compressed database. To convert the MDED distance to a similarity measure, we create again a RBF-like kernel as $k_{\mathrm{MDED}}(\boldsymbol{X},\boldsymbol{Y}) = \exp(-\mathrm{MDED}(\boldsymbol{X},\boldsymbol{Y})/\sigma)$, where the bandwidth parameter σ is set to the median of $\mathrm{MDED}(\boldsymbol{X},\boldsymbol{Y})$.

4 Evaluation of Forecasting Performance

We evaluate the forecasting performance of the proposed methodology to capture the linear inter-dependencies among multiple time series. We base our experiments on SVR using different multivariate dynamic kernels, namely k_{GA}, k_{VAR} and k_{MDED}. We compare also against the VAR model, a standard in econometrics, although it does not allow to integrate mixed-frequency information from markets. The goal is to forecast the next month return of Standard and Poor's 500 Index (S&P500) by incorporating past information plus three *exogenous* predictors (hence $P = 4$): the volatility index (VIX), the yield of the U.S. 10-year treasury bond (US10Yr) and the price of cooper 3-month future contract (LME3m). All models were tested along three different time windows so as to evaluate the effect of distinct market regimes in prediction accuracy, based on compressed daily historical prices from January 2006 to December 2014.

The output variable of the model is the next month log-return of S&P500, R_{t+1}. We use the log-return because it has better statistical properties than price, as stationarity and ergodicity [9]. The inputs are constructed on a daily basis to capture temporal patterns of different scale on S&P500, VIX, US10yr and LME3m using the $\mathrm{ROC}_{t,n} = \ln(x_t) - \ln(x_{t-n})$ function for n days on day t.

For the i-th time series ($i = 1, \ldots, 4$), we derive a vector of several rates of changes on each day t, incorporating the time series at $n \in \{20, 40, 60, 100, 140\}$, allowing to capture temporal trend shifts of financial markets when analyzed on a monthly basis. Then the input features for day t take the form $x_t = [x_t^1, x_t^2, x_t^3, x_t^4]$, where $x_t^i = [\mathrm{ROC}^i_{t,20}, \mathrm{ROC}^i_{t,40}, \mathrm{ROC}^i_{t,60}, \mathrm{ROC}^i_{t,100}, \mathrm{ROC}^i_{t,140}]$.

Methodology and Parameter Selection. In the ν-SVR model, ν is constrained to the interval $(0, 1]$. We optimize it in the set $\{0.1, 0.2, \ldots, 1\}$. For the possible choices of C, we follow the analytic approach proposed by [10], which advocates parameter selection directly from the training data. Considering a standard SVR solution, a reasonable value for C can be roughly equal to the range of training output values. However, besides forecasting with a value $C = \mathrm{range}(R_{t+1})$, we also tried values in the set $\left\{\mathrm{range}(R_{t+1}) \cdot \{0.8, 0.9, 1, 1.1, 1.2\}\right\}$.

To find the optimal parameters ν and C and the fitted models we use the methodology of [11], combining rolling windows with "training-validation-testing" blocks. Despite being a standard practice in financial applications, rolling windows are uncommon in the machine learning literature. An in-sample period of 6 months was decided to train the model to make predictions for the next month. The proposed methodology to predict the market performance is a multi-step procedure. First, we train the models on 6 months (the training set); second, we apply the resulting models on the next two months (the validation set) and select the values of parameters that achieve the minimum mean squared error; and third, we combine the last 4 months of the training set and the 2 months of the validation set into a new set, called "true training set", and train the final model using the selected values of ν and C. Finally, we apply the model on the next month (the test set) and record its performance. We then move one month forward, repeating the same procedure for the whole period.

Performance Metrics. A number of measures have been used in the literature to compare the forecasting accuracy of different models. Popular measures –such as the mean squared error– are not invariant to scaling. We use here the *mean absolute scaled error* (MASE), which scales the measured error using the mean absolute error of a naive forecast:

$$\text{MASE} = \text{mean}\left|\frac{e_t}{\frac{1}{n-1}\sum_{t=2}^{n}|Y_t - Y_{t-1}|}\right|$$

where Y_t denotes the observation at time $t \in \{1, \ldots, n\}$, F_t is the model forecast and $e_t = Y_t - F_t$ is the forecast error. A MASE smaller than 1 indicates that forecasting performance is better than a naive forecast. In addition, we compute the accuracy or hit rate (HITS) –which should be maximized– as $\text{HITS} = \text{mean}|\{F_t \mid (Y_t - Y_{t-1}) \cdot (F_t - F_{t-1}) > 0,\ t = 1, \ldots, n\}|$.

Empirical Results. Table 1 shows the MASE and HITS results we obtain from using the multivariate dynamic kernels within the SVR framework; we also report the performance of the VAR model. All results are presented both for the whole database period and for balanced time windows, so as to capture the performance of kernels across different market regimes.

	MASE				
	Naive	k_{GA}	k_{VAR}	k_{MDED}	VAR
2006–08	1.000	0.783	0.795	0.769	1.151
2009–11	1.000	0.896	0.850	0.846	1.130
2012–14	1.000	0.728	0.712	0.733	1.568
Total	1.000	0.819	0.798	0.794	1.246

	HITS				
	Naive	k_{GA}	k_{VAR}	k_{MDED}	VAR
2006–08	0.657	0.778	0.750	0.722	0.639
2009–11	0.500	0.583	0.528	0.556	0.611
2012–14	0.583	0.722	0.722	0.722	0.417
Total	0.579	0.694	0.667	0.667	0.556

Table 1. MASE (left) and HITS (right) of the Multivariate Dynamic Kernels.

The results clearly show the ability of SVR with multivariate dynamic kernels to forecast the financial market. The kernels are able to achieve overall mean absolute squared errors of about 80%, accounting for an improvement of 20% in performance with respect to the naive forecast. The most troublesome period for forecasting is between 2009 and 2011, when financial markets underwent profound trend shifts due to the world crisis. The VAR model is outperformed both by the naive forecast and the multivariate dynamic kernels in all periods. There are many possible explanations, the most important in our opinion is that it is based on strong assumptions (linearity, stationarity, etc.) that do not fit well to financial markets, particularly when working with small data sets.

In predicting market trends, the multivariate dynamic kernels reach a hit rate of up to 70% over the whole period, compared to a hit rate of 58% for the naive forecast. This is particularly remarkable because the hit rate is very used in algorithmic trading by signaling actions upon predicted market trend shifts.

As we demonstrate, the multivariate dynamic kernels lead to significant improvements in prediction accuracy and better performance than the naive forecast along different market regimes. They also outperform the VAR model in nearly all periods. The proposed MDED kernel and the modified version of the VAR kernel display a performance similar to that of the global alignment kernel, which is the state-of-the-art similarity measure in the literature for variable-length time series. In fact, when analyzed in each period, we can note there is no decisive winner among the kernels. The CPU times[1] (in seconds) are 156, 143, 33 and 0.7, respectively, for $k_{\mathrm{GA}}, k_{\mathrm{VAR}}, k_{\mathrm{MDED}}$ and VAR, indicating the computational efficiency of the proposed MDED kernel. The VAR model is the fastest forecaster but it is not capable of performing better than the naive forecast.

An Experiment in Trading. We now apply the method to forecast the financial market and compare performance against the *buy-and-hold* strategy, widely used as a benchmark in financial research. We follow the approach of [11] defining a simple investing strategy: let $\hat{f}_{t+1}$ be the forecasted S&P500 next month log-return; if $\hat{f}_{t+1} \geq 0$, we buy at the closing price on month t; otherwise, we short it. Then the log-return $\hat{R}_{t+1}$, associated with our strategy, can be computed as:

$$\hat{R}_{t+1} = \begin{cases} |R_{t+1}| & \text{if } \ R_{t+1} \cdot \hat{f}_{t+1} \geq 0 \\ -|R_{t+1}| & \text{otherwise.} \end{cases}$$

	B&H	0 bp.	30 bp.	50 bp.
k_{GA}				
Total cum. (%)	50.04	145.64	133.62	125.59
Mean (%)	5.56	16.18	14.85	13.96
Stdev (%)	15.54	14.90	14.94	14.97
Sharpe ratio	0.36	1.09	0.99	0.93
k_{VAR}				
Total cum. (%)	50.04	123.78	113.86	107.24
Mean (%)	5.56	13.75	12.65	11.92
Stdev (%)	15.54	15.11	15.16	15.21
Sharpe ratio	0.36	0.91	0.83	0.78

	0 bp.	30 bp.	50 bp.
k_{MDED}			
Total cum. (%)	131.99	122.08	115.45
Mean (%)	14.67	13.56	12.83
Stdev (%)	15.03	15.10	15.15
Sharpe ratio	0.98	0.90	0.85
VAR			
Total cum. (%)	67.19	45.86	31.60
Mean (%)	7.47	5.10	3.51
Stdev (%)	15.47	15.57	15.65
Sharpe ratio	0.48	0.33	0.22

Table 2. Statistics for SVR timing rotation strategies with transactions costs.

[1] Laptop with 4 GB of RAM and Intel Core i5 processor running at 2.5 GHz.

The predicted performance of the financial market for the next month is thus used on a timing rotation strategy. A positive prediction turns into a "buy signal", in which an Exchange Traded Fund (ETF) tracking the S&P500 index is bought, whereas a negative one results in short-selling the ETF. We have included different levels of transaction costs that take off some basis points or bp (equal to a 0.01%) of the capital for each trade. Table 2 shows a summary of the investment strategy performance with different kernels in the period between January 2006 and December 2014. What strikes at first sight, is that all kernels invariably yield better results than the buy-and-hold (B&H) strategy.

Under the assumption of zero transaction cost, the average annual log-return of multivariate dynamic kernels ranges between 2.47 and 2.91 times the B&H strategy. Indeed, the GA kernel achieves an annual mean return of 16.18%, the VAR kernel 13.75% and the MDED kernel 14.67%, compared to the buy-and-hold strategy of 5.56%. Combining these results with the standard deviations yields improvements of more than 2.5 times in the Sharpe ratio. When adding conservative transaction costs of 30 bp. and 50 bp. the results remained superior to the buy-and-hold strategy. The VAR model modestly outperforms the passive strategy and only when transaction costs are smaller than 30 bp.

The MDED kernel might then be effectively applied when considering high-frequency time series for horizons of minutes or seconds. All the kernels can play a major role in market risk management by the approximation of quantiles for a certain distribution like, for example, in the value-at-risk (VaR) along with the incorporation of the latest intra-day market developments.

References

1. Fama, E. F.: Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2):383–417 (1970)
2. Duan, W., Stanley, H.: Cross-correlation and the predictability of financial return series. Physica A: Statistical Mechanics and its Applications, 390(2):290–296 (2011)
3. Kim, K.-J.: Financial time series forecasting using support vector machines. Neurocomputing, 55(1):307–319 (2003)
4. Tay, F. E., Cao, L.: Application of support vector machines in financial time series forecasting. Omega, 29(4):309–317 (2001)
5. Smola, A. J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing, 14(3):199–222 (2004)
6. Sakoe, H., Chiba, S.: A similarity evaluation of speech patterns by dynamic programming. In Meeting of Inst. of Electr. Communs. Engineers of Japan (1970)
7. Cuturi, M., Vert, J.-P., Birkenes, Ø., Matsui, T.: A kernel for time series based on global alignments. In IEEE Int. Conf. ICASSP 2007, pages II–413. IEEE (2007)
8. Cuturi, M., Doucet, A.: Autoregressive kernels for time series. Technical Report arXiv:1101.0673 (2011)
9. Tsay, R.: Analysis of Financial Time Series, Wiley (2010)
10. Mattera, D., Haykin, S.: Support vector machines for dynamic reconstruction of a chaotic system. In Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, pages 211–241. MIT Press (1999)
11. Wang, L., Zhu, J.: Financial market forecasting using a two-step kernel method for the support vector regression. Annals of Operations Res., 174(1):103–120 (2010)